Endian and Related Environment Variables or Compiler Options

Category: Porting & Developing Applications

DRAFT

This article is being reviewed for completeness and technical accuracy.

Intel Fortran expects numeric data, both integer and floating-point data, to be in native little endian order, in which the least-significant, right-most zero bit (bit 0) or byte has a lower address than the most-significant, left-most bit (or byte).

If your program needs to read or write unformatted data files that are not in little endian order, you can use one of the six methods (listed in the order of precedence) provided by Intel below.

1. Set an environment variable for a specific unit number before the file is opened. The environment variable is named **FORT_CONVERTn**, where n is the unit number. For example:

   ```
   setenv FORT_CONVERT28 BIG_ENDIAN
   ```

   No source code modification or recompilation is needed.

2. Set an environment variable for a specific file name extension before the file is opened. The environment variable is named **FORT_CONVERT.ext** or **FORT_CONVERT_ext**, where ext is the file name extension (suffix). The following example specifies that a file with an extension of .dat is in big endian format:

   ```
   setenv FORT_CONVERT.DAT BIG_ENDIAN
   ```

   Some Linux command shells may not accept a dot (.) for environment variable names. In that case, use FORT_CONVERT_ext instead.

   No source code modification or recompilation is needed.

3. Set an environment variable for a set of units before any files are opened. The environment variable is named **F_UFMTENDIAN**.

   Syntax:

Csh: setenv F_UFMTENDIAN MODE;EXCEPTION

Sh : export F_UFMTENDIAN=MODE;EXCEPTION

MODE = big | little

EXCEPTION = big:ULIST | little:ULIST | ULIST

ULIST = U | ULIST,U

U = decimal | decimal-decimal

MODE defines the current format of the data, represented in the files; it can be omitted. The keyword "little" means that the data have little- endian format and will not be converted. For IA-32 systems, this keyword is a default. The keyword "big" means that the data have big endian format and will be converted. This keyword may be omitted together with the colon.

EXCEPTION is intended to define the list of exclusions for MODE; it can be omitted. EXCEPTION keyword (little or big) defines data format in the files that are connected to the units from the EXCEPTION list. This value overrides MODE value for the units listed.

Each list member U is a simple unit number or a number of units. The number of list members is limited to 64. decimal is a non-negative decimal number less than 2**32.

The environment variable value should be enclosed in quotes if the semicolon is present.

Converted data should have basic data types, or arrays of basic data types. Derived data types are disabled.

Example:

- setenv F_UFMTENDIAN big

  All input/output operations perform conversion from big-endian to little-endian on READ and from little-endian to big-endian on WRITE.

- setenv F_UFMTENDIAN "little;big:10,20"

  or setenv F_UFMTENDIAN big:10,20

  or setenv F_UFMTENDIAN 10,20

  In this case, only on unit numbers 10 and 20 the input/output operations perform big-little endian conversion.

- setenv F_UFMTENDIAN "big;little:8"

  In this case, on unit number 8 no conversion operation occurs. On all other units, the input/output operations perform big-little endian conversion.
- setenv F_UFMTENDIAN 10-20

  Define 10, 11, 12, ...19, 20 units for conversion purposes; on these units, the input/output operations perform big-little endian conversion.

4. Specify the **CONVERT** keyword in the OPEN statement for a specific unit number. Note that a hard-coded OPEN statement CONVERT keyword value cannot be changed after compile time. The following OPEN statement specifies that the file graph3.dat is in VAXD unformatted format:

   ```
   OPEN (CONVERT='VAXD', FILE='graph3.dat', FORM='UNFORMATTED',
   UNIT=15)
   ```
5. Compile the program with an **OPTIONS** statement that specifies the CONVERT=keyword qualifier. This method affects all unit numbers using unformatted data specified by the program. For example, to use VAX F_floating and G_floating as the unformatted file format, specify the following OPTIONS statement:

   ```
   OPTIONS /CONVERT=VAXG
   ```
6. Compile the program with the command-line **-convert keyword** option, which affects all unit numbers that use unformatted data specified by the program. For example, the following command compiles program file.for to use VAXD floating-point data for all unit numbers:

   ```
   ifort file.for –o vconvert.exe –convert vaxd
   ```

In addition, if the record length of your unformatted data is in byte units (Intel Fortran default is in word units), use the **-assume byterecl** compiler option when compiling your source code.

Article ID: 206
Last updated: 24 May, 2011
Computing at NAS -> Porting & Developing Applications -> Endian and Related Environment Variables or Compiler Options
http://www.nas.nasa.gov/hecc/support/kb/entry/206/?ajax=1